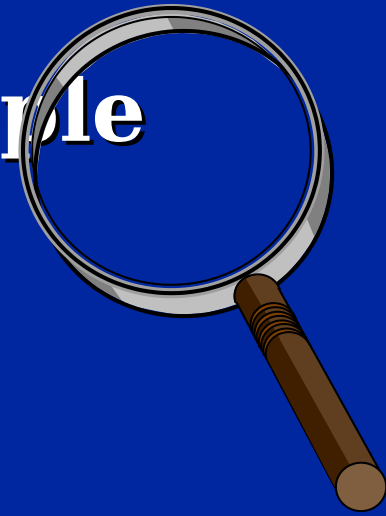




WindowsTM User Interface “Chicago” and “Cairo” End- User Joe Belfiore Personal Systems Steve Madigan Advanced Systems Microsoft Corporation

Windows User Interface Design Focus

- ◆ Focus on user *experience*
- ◆ Understandable *objects*
- ◆ System and application *integration*
- ◆ Support for the *tasks* people do
- ◆ Where we're headed
- ◆ Design process



Windows User Interface:

- ◆ The heritage of an application-based model
- ◆ “Windows” hovering on wallpaper
- ◆ Lots of middle management
 - Program Manager and File Manager
 - ...and “thing” Managers and Control Panel
- ◆ Independent applications

...too much computer
junk in the way of
getting things done

Windows User Interface:

- ◆ **The system defines a *user environment***
 - How things are created
 - How things are organized
 - How things are found
- ◆ ***Objects* make the environment useful**
- ◆ ***Applications* create capabilities**
 - To do new things
 - To work with new objects
- ◆ ***Tasks* cross the boundaries that objects create**
- ◆ ***Information* is easily found and used**

Evolution Of The UI Design Process

- ◆ Broader set of disciplines required in the design team
- ◆ Greater emphasis on graphic design
- ◆ More user involvement, earlier and throughout the design process
 - Observation of users in their workplace
 - Interviews with educators
 - Extensive usability testing
- ◆ Greater focus on getting underlying technology in line

Windows User Interface

- ◆ Renewed focus on user *experience*
- ◆ Understandable *objects*
- ◆ “Are my *what* integrated?”
- ◆ Support for the *tasks* people do

Striving for a user *environment*
that is
understandable, seamless, and
supportive
of the many ways people work

Rollout Of The Windows

“Chicago” in '94 “Cairo” in '95 End-User Environment

- ◆ New look and feel debut
- ◆ New focus on supporting end-user *tasks*
- ◆ OLE integration
- ◆ Desktop integration
- ◆ Simplified network UI

- ◆ Continuing look and feel
- ◆ Focus on people working together
- ◆ OLE-based, fully extensible system user interface
- ◆ OLE forms and controls
- ◆ Query

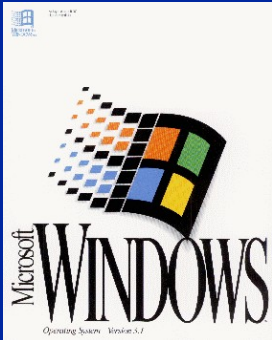
What's The Windows “*Chicago*” UI About?

- ◆ Focus on *user experience*
 - Object-oriented, document-centric
 - Includes a task-based component
- ◆ Focus on *design*
 - New visual appearance
- ◆ Provide “*helpers*” so applications can maintain and extend the metaphor
 - Shell extensibility: seamless integration

Who's It For?

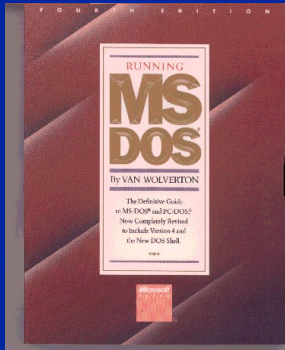
Windows™ 3.1-based users

- ◆ More efficient access to common tasks
- ◆ New features
- ◆ Conceptual model is easier to understand



New GUI users

- ◆ Entry-level tasks are discoverable
- ◆ Conceptual obstacles (hierarchy, window management) are reduced
- ◆ Mechanical obstacles (double-click) are removed



Central Elements

- ◆ *Desktop*: where the big containers are
- ◆ *Folders*: so users can organize their belongings
- ◆ *Documents*: not hidden inside some application you need to run...
- ◆ *Desktop toolbar*: quick access to common tasks
 - Start a program
 - Open a document, etc.

What this means to you

***Build a great application for
“Chicago”!***

- ◆ **Exploit our new usage model:
Be document-centric and
object-oriented**
- ◆ **Use our controls, dialogs, and
extensibility mechanisms**
- ◆ **Put your application in front of
LOTS of users**

“Chicago” UI Points of Light

1. Support Long Filenames.
2. Support UNC pathnames.
3. Make sure your documents/data files are accurately displayed and used in the shell -- icons, no .3 extensions displayed.
4. Support drag & drop and other transfers consistently and extensively.
5. Use the common dialog boxes, especially File Open/Save As.
6. Maintain a consistent user interface and object paradigm between your app and the shell -- context menus, doc-centric title bar, right-drag.
7. Extend the shell's ability to provide general information about your files (*more* OLE 2.0).
8. Support “Chicago”-style Help in your application.
9. Follow “Chicago’s” application setup guidelines to make your app properly visible in the shell.
10. Be careful about multiple instances of your app being started *too easily* at the same time.

The Path From Windows “*Chicago*” UI To

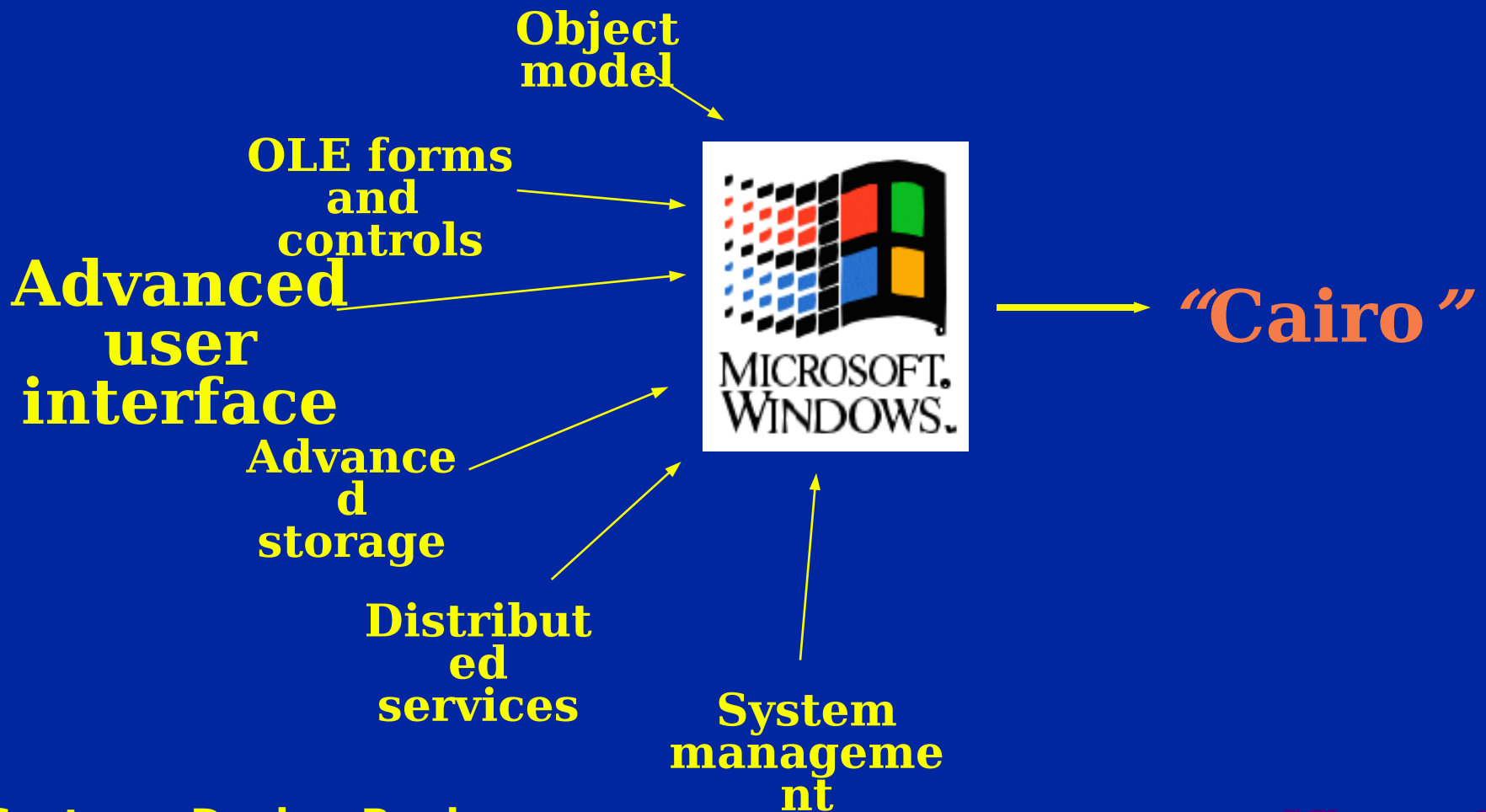
Windows NTTM “*Cairo*” adds Common UI elements:

- ◆ Visuals
- ◆ Desktop
- ◆ Explorer
- ◆ Property sheets
- ◆ Context menus
- ◆ Views
- ◆ Scraps

- ◆ Multitude of new features and extensions
- ◆ New folder views
- ◆ Improved UI for information access
- ◆ Full system UI extensibility
- ◆ OLE Automation support

- # Windows NT Cairo User Interface Goals
- ◆ Support the work people do... together
 - ◆ *Information*
 - Creation, sharing, using, and finding
 - Regardless of what or where it is
 - ◆ Even simpler and more natural for new users
 - ◆ One model for manipulating all objects
 - ◆ Even more fun and satisfying to use
 - ◆ Innovation within an evolutionary framework

The Opportunity: Technology Integration



Windows NT “Cairo” User

◆ System/application integration

- Fully extensible system UI
- OLE Automation support

◆ *Information*

- Finding
- Using
- Creating

An Integrated System, Integrated Applications

- ◆ **Reuse of shell UI components**
- ◆ **Enhancing the capabilities of the shell through OLE-based extensibility**
 - **Finding and viewing information in the Explorer**
 - **Describing objects through properties**
 - **Creating UI elements based on OLE forms**
- ◆ **Supporting OLE Automation in the shell**
 - **The groundwork for automating tasks**

Finding Things

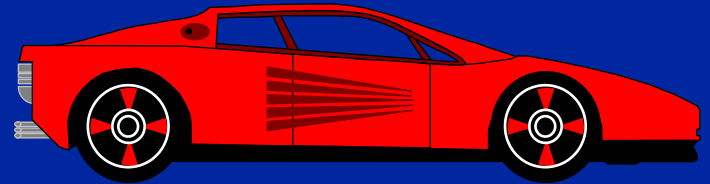
- ◆ Ask for what you want, the system brings it to you
 - All kinds of things
 - All kinds of places
- ◆ System support for all kinds of finding activities
 - “Searching”
 - “Researching”
 - “Browsing”
 - ...then keeping hold of, or using, what you find
- ◆ An embedded part of work processes

Cruising The Information Highway

“How nice the road is depends on what you drive”

◆ The Information “Highway”

- Local storage**
- Network storage**
- Directory services**
- Databases**
- Many different information sources**



Windows User Interface

◆ Windows NT “Cairo” *A user experience*

- A simple, consistent model for objects and information

◆ *Extensibility*

- Of system UI components through OLE

◆ *Seamlessly integrated*

- System and applications

◆ *Finding information so FAST*

- That it changes the way people